

شیوه طراحی واکنش گرا بودن

تعریف طراحی واکنش گرا (Responsive Design)

در دنیای امروزی، کاربران از دستگاه‌های مختلفی برای دسترسی به اینترنت استفاده می‌کنند، از گوشی‌های هوشمند گرفته تا تبلت‌ها، لپ‌تاپ‌ها و مانیتورهای بزرگ. هر یک از این دستگاه‌ها دارای اندازه صفحه، رزولوشن و نسبت تصویری متفاوتی هستند. بنابراین، طراحی وبسایت‌ها به گونه‌ای که در تمامی این دستگاه‌ها به درستی نمایش داده شوند، امری ضروری است.

طراحی واکنش گرا (Responsive Web Design - RWD) روشی در طراحی و توسعه وب است که هدف آن ایجاد صفحاتی است که به‌طور خودکار خود را با اندازه صفحه‌نمایش و نوع دستگاه کاربر سازگار می‌کنند. این نوع طراحی باعث می‌شود که محتوای سایت در تمامی دستگاه‌ها (مانند موبایل، تبلت، لپ‌تاپ و دسکتاپ) خوانا، کاربرپسند و بهینه باشد، و دیگر نیاز به ایجاد نسخه‌های جداگانه برای هر دستگاه نیست.

دلایل اهمیت طراحی واکنش گرا

۱. **افزایش دسترسی پذیری:** کاربران از دستگاه‌های مختلفی برای دسترسی به وبسایت‌ها استفاده می‌کنند. طراحی واکنش گرا اطمینان حاصل می‌کند که همه کاربران تجربه‌ای مناسب خواهند داشت.

۲. **بهبود تجربه کاربری (UX):** طراحی واکنش گرا باعث می‌شود که کاربران بتوانند به راحتی از وبسایت استفاده کنند، بدون اینکه نیاز به زوم کردن، اسکرول افقی یا تلاش برای کلیک کردن روی لینک‌های کوچک داشته باشند. این مسئله به ویژه در دستگاه‌های لمسی اهمیت زیادی دارد، زیرا دکمه‌ها و لینک‌ها باید به اندازه‌ای باشند که کاربران بتوانند به راحتی روی آن‌ها کلیک کنند.

۳. **سئو و بهینه‌سازی موتورهای جستجو (SEO):** گوگل و سایر موتورهای جستجو، وبسایت‌هایی را که طراحی واکنش گرا دارند، در رتبه‌بندی نتایج جستجو بالاتر نمایش می‌دهند. از آنجایی که طراحی واکنش گرا نیاز به نسخه‌های جداگانه برای موبایل و دسکتاپ ندارد، از ایجاد محتوای تکراری (Duplicate Content) جلوگیری می‌کند و مدیریت وبسایت را ساده‌تر می‌سازد.

۴. **کاهش هزینه و زمان توسعه:** به جای ایجاد نسخه‌های جداگانه برای موبایل و دسکتاپ، یک نسخه طراحی می‌شود که در همه دستگاه‌ها به درستی نمایش داده می‌شود.

نحوه عملکرد طراحی واکنش گرا

طراحی واکنش گرا بر اساس سه اصل کلیدی پیاده سازی می شود:

۱. استفاده از طراحی سیال (**Fluid Layouts**): به جای استفاده از اندازه های ثابت (px)، از واحدهای نسبی مانند درصد (/) و vw/vh (عرض و ارتفاع متناسب با صفحه) استفاده می شود تا المان های صفحه متناسب با اندازه صفحه تغییر کنند.

۲. استفاده از **Media Queries**: این قابلیت در CSS به ما اجازه می دهد که بر اساس اندازه صفحه، استایل های متفاوتی اعمال کنیم تا محتوا متناسب با دستگاه نمایش داده شود.

۳. طراحی مبتنی بر **Grid** و **Flexbox**: استفاده از CSS Grid و Flexbox به جای روش های قدیمی مانند استفاده از جدول ها، باعث می شود که چیدمان صفحه انعطاف پذیرتر و قابل تنظیم تر باشد.

۴. استفاده از تصاویر و رسانه های واکنش گرا: تصاویر و ویدئوها باید به گونه ای طراحی شوند که متناسب با عرض صفحه تغییر اندازه دهند. (max-width: 100%)

ما در این پروژه براساس نیاز از هر سه اصل طراحی واکنش گرا استفاده کرده ایم:

- طراحی سیال و مبتنی بر Flexbox:

در این بخش صفحه اصلی flex و همچنین طول صفحه با پارامتر vh اندازه داده شده است که انعطاف پذیری صفحه را بالا برده است. این بخش برای تمامی صفحات مشترک است.

```
body {  
  font-family: 'Work Sans', sans-serif;  
  background: linear-gradient(to bottom right, #A888B5, #FFFAF1);  
  margin: 0;  
  padding: 0;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  height: 100vh;  
}
```

- طراحی مبتنی بر Flexbox و max-width:

در این قسمت بخش اصلی صفحه با طول و عرض ۱۰۰٪ تعریف شده است که در هر سایز از صفحات (موبایل، تبلت، لپ‌تاپ و دسکتاپ) ۱۰۰٪ صفحه را در بر گیرد.

```
.main-container {  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  width: 100%;  
  height: 100%;  
}
```

login.css

```
.search-container {  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  width: 100%;  
  max-width: 600px;  
  background: white;  
  border-radius: 50px;  
  padding: 8px;  
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);  
  margin: 20px auto;  
}
```

home.css

```
.profile {  
  width: 35px;  
  height: 35px;  
  border-radius: 50%;  
  object-fit: cover;  
}
```

crm.css

```
.button {  
  width: 100%;  
  padding: 12px;  
  background-color: #9a7bbd;  
  border: none;  
  color: white;  
  font-size: 16px;  
  font-weight: 600;  
  border-radius: 8px;  
  cursor: pointer;  
  transition: 0.3s;
```

```
}
```

register.css

- استفاده از Media Queries و max-width :

```
@media (max-width: 400px) {  
  .form-container {  
    width: 90%;  
  }  
}
```

login.css

```
@media (min-width: 668px) {  
  .button-74-logout {  
    min-width: 100px;  
    padding: 0 20px;  
  }  
}
```

crm.css

```
@media screen and (max-width: 600px) {  
  .member-info {  
    flex-direction: column;  
    align-items: center;  
    text-align: center;  
    padding: 8px;  
  }  
  
  .profile {  
    width: 30px;  
    height: 30px;  
  }  
  
  .words-count {  
    margin-top: 5px;  
    font-size: 11px;  
    padding: 4px 6px;  
  }  
}
```

crm.css

```
@media (min-width: 768px) {  
  .button-74 {  
    min-width: 120px;  
    padding: 0 25px;  
  }  
}
```

```
}  
}
```